
Subject: Re: IDL Style (was Re: IDL 5.2 GUI Builder Tutorial ?)

Posted by [Michael Asten](#) on Tue, 01 Jun 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

>
>>
>> Mike -
>>
>> What exactly is so bad about this?
>
> Hi Gwyn -
>
> No clue what base0, base1, label1, etc. indicate, and they're repeated
> throughout the example, and throughout dozens of files in the project.
> Something descriptive is really useful, especially when you're dealing
> with a lot of files, all of which look identical, and all of which have no
> reasonable amount of comments. Literally, there is about a 100:1 ratio for
> code:comments in the project. It's the kind of code slapped together in an
> afternoon with no consideration for the people to follow along afterwards
> and make changes, enhancements, deletions, etc.

Hi guys,
arent we confusing a couple of different issues here?

Firstly, thanks to Mike S. for the earlier posting on the style guide. I like it. I tend to use a default fortran-style naming convention, but the additional rigor suggested by Mike makes sense. In hindsight it would have saved me a few "what did I mean when I wrote this 3 months ago" experiences.

Secondly, regarding undocumented widget code generated by WIDED or GUIBUILDER, I don't accept the argument that poor documentation is an argument against their use. Their use saves a great deal of programming time for some of us, and before we give away, (or archive for ourselves!) the code, it is commonsense professional practice to add blocks of comment code. This will be "what the widget does" rather than individual lines of "how the code does it". Mikes angst with the busy person who failed to document before putting code into the public domain, is justifiable, but keep in mind that a lot more tax dollars might have been spent if the programmer had worked from a lower level.

I believe that arguments against the use of guibuilders are akin to arguments against the use of 4GLs generally (amazing the number of professional programmers who think that programming in any language higher-level than c++ is for wimps).

One of the beauties of machine generated widget code, apart from initial time savings, is that it is predictable and systematic and requires less documentation than a block of code written by a person with a personal style. However one has to resist the temptation to hack the generated code to pieces. I think the guibuilder is a good attempt in the right direction, since it encourages

separation of event handling code from gui code. I also find that the much-maligned WIDED is superior when it comes to making a gui to handle text fields (especially real numbers) . I'd love to see rsi produce a clear tutorial on how to get the most out of the guibuilder, to the same standard as say David Fannings tutorials on how to avoid it.

Regards,
Michael Asten
